

# Have Your Cake and Eat it Too? Managing Knowledge in Hybrid Agile-Traditional Development Projects

*Full Paper*

**W. Alec Cram**  
Bentley University  
[wcram@bentley.edu](mailto:wcram@bentley.edu)

**Marco Marabelli**  
Bentley University  
[mmarabelli@bentley.edu](mailto:mmarabelli@bentley.edu)

## Abstract

This research examines how knowledge management elements of systems development projects are adapted to correspond with the increasing use of agile practices. Using a single, longitudinal case study, we draw on interview data from an initial CRM implementation that used a traditional approach, followed by a second project phase employing a hybrid agile-traditional approach. Oriented using the concept of ambidexterity, our findings suggest that some knowledge management elements remained traditional, others became strongly agile, and some integrated both approaches together in a hybrid manner. Our study aids practitioners by providing insights into the opportunities and pitfalls of managing knowledge within hybrid development projects. For researchers, this paper applies the concept of ambidexterity in an agile development context as it relates to knowledge management. Our findings extend the literature on the incremental trade-offs that companies face when attempting to simultaneously explore and exploit two development approaches.

## Keywords

Agile development, traditional development, knowledge management, ambidexterity

## Introduction

The increasing popularity of agile methods over the last decade has significantly altered the systems development process within many organizations, where a transition is under way from a structured, waterfall approach to an increasingly flexible, iterative approach (Dingsoyr et al. 2012; Nerur et al. 2005; Vinekar et al. 2006). However, past information systems (IS) research suggests that organizations commonly stop short of fully embracing all aspects of agile, preferring instead to customize their development approach to integrate both traditional and agile practices together (Boehm and Turner 2003; Port and Bui 2009; Vinekar et al. 2006). For example, a project may integrate traditional-based practices such as up-front planning and budgeting alongside agile-based practices such as pair programming and shared code ownership. By tailoring a development approach to complement the unique characteristics of a process, customer, or organization, development performance can thus be enhanced (Cao et al. 2009; Fitzgerald et al. 2006).

Among the many features that distinguish a traditional development approach from an agile approach is the role of knowledge (Cabral et al. 2009). For example, while agile relies largely on tacit forms of knowledge (Polanyi 1962) shared through practices such as stand-up meetings and pair programming, traditional methods draw instead on explicit knowledge through the use of documents and formal plans (Vinekar et al. 2006). As a result, systems development projects that employ a hybrid agile-traditional approach are faced with the conflicting demands of knowledge management that align with an agile approach, a traditional approach, or some combination of the two. Because effective knowledge management has been shown to be a key driver of project performance (Chau et al. 2003; Melnik and Maurer 2004), it is important to understand how organizations reconcile these competing options in practice. In this paper, we look at the trade-offs between agile and traditional development approaches using the theoretical lens of ambidexterity (Gibson and Birkinshaw 2004; Tushman and O'Reilly 1996).

Specifically, we consider how development projects transition away from the exclusive use of a traditional approach and develop ambidextrous capabilities through the simultaneous adoption of agile and traditional approaches.

In this paper we focus on one specific element that distinguishes traditional from agile development: knowledge management. We seek to understand if organizations using a hybrid approach are forced to choose a tactic to manage knowledge that is associated with a single development approach or if projects can integrate diverse knowledge management elements in an attempt to reap the benefits of both approaches, while avoiding the drawbacks; in effect, having your cake *and* eating it too. Specifically, we pose the following research question: *how do the knowledge management elements of systems development approaches change as agile practices are incrementally adopted alongside traditional practices?* We draw on a longitudinal, single site case study examining a customer relationship management (CRM) system implementation that employed a traditional approach in the first phase of the project, followed by a hybrid agile-traditional approach in the second phase.

The contribution of this paper is twofold. Theoretically, our study applies the concept of ambidexterity in an agile development context as it relates to knowledge management. The resulting findings provide insights into the trade-offs that companies make in attempting to simultaneously pursue two development approaches. From a practical perspective, our research aids in developing an enhanced understanding into the challenges of effectively mobilizing knowledge within hybrid projects. By understanding one company's path from a traditional approach to a hybrid agile-traditional approach, these insights can aid managers in recognizing the opportunities and pitfalls inherent in adopting ambidextrous systems development competencies.

The remainder of the paper is as follows. In the next section, we provide an overview of the concept of ambidexterity, a discussion of agile and traditional development approaches, and the role of knowledge management in systems development projects. Next, we outline our methodology, including details of the data collection and analysis. The results are then presented and our findings are discussed. We conclude by presenting a series of future research opportunities.

## **Background and Theoretical Base**

This study applies an ambidexterity lens to examine knowledge management within systems development projects that employ a hybrid agile-traditional approach. We first introduce the concept of ambidexterity and then discuss it in an agile/traditional context, including the potentially conflicting elements related to knowledge management.

### ***Ambidexterity in Hybrid Agile-Traditional Development Projects***

At its broadest level, the concept of ambidexterity refers to an organization's pursuit of two disparate objectives simultaneously (Gibson and Birkinshaw 2004), though more specific definitions position ambidexterity as the balance between exploiting the use of existing knowledge, while also exploring new knowledge in order to remedy deficiencies (Turner et al. 2013). In order to achieve long-term success, ambidexterity enables continual adaptation over time in response to both small and large changes in strategy, culture, and structure (Raisch et al. 2009; Tushman and O'Reilly 1996).

Widely used within the organizational, innovation and technology management, marketing, and strategy literature, ambidexterity has been applied at the company, group, and individual level (Raisch et al. 2009; Turner et al. 2013). By considering how new opportunities can be explored, while existing opportunities are exploited, the IS literature addresses topics such as the conflicting demands associated with internal and external sourcing (Rothaermel and Alexandre 2009), formal and informal controls (Tiwana 2010), open innovation (Stoetzel and Wiener 2013), and CRM technology (Li et al. 2013).

Ambidexterity is commonly positioned as either structural, where different subunits of an organization independently pursue alternative approaches, or contextual, where the conflicting demands are addressed within the same business unit (Gibson and Birkinshaw 2004; Marabelli et al. 2012). Over the past decade, the popularity of agile development has given rise to both structural and contextual forms of ambidexterity, as companies seek to explore the speed and flexibility of agile methods while continuing to exploit traditional (e.g. waterfall) methods (Ramesh et al. 2012; Vinekar et al. 2006). In contrast to these

traditional methods, which rely on sequential activities, extensive planning and documentation, role specialization, and hierarchical oversight, agile refers to a family of development methods (e.g. Scrum, XP) that advocate for a customer-oriented approach relying on techniques such as minimal documentation, regular delivery of working code, and collaboration with users (Beck et al. 2001; Dyba and Dingsoyr 2008; Nerur and Balijepally 2007; Nerur et al. 2005).

However, many companies have shown reluctance to completely abandon a traditional approach in favor of agile and are instead adopting elements of both (Cao et al. 2009; Fitzgerald et al. 2006; Nerur et al. 2005). In some cases, companies pursue structural ambidexterity by developing independent capabilities in both approaches and then select only one for use on a development project (Vinekar et al. 2006), while other firms pursue contextual ambidexterity by integrating practices of both approaches together simultaneously (Ramesh et al. 2012). Past research has focused on the explorative opportunities presented by agile methods, the exploitive opportunities of traditional methods, as well as the potential conflicts that emerge when using a hybrid, integrated approach (Vinekar et al. 2006).

Projects pursuing contextual ambidexterity employ hybrid development approaches that consolidate traditional practices (e.g. extensive planning and documentation) with agile practices (e.g. pair programming, story cards). This offers elements of an approach that is proven and reliable (i.e. traditional), while simultaneously developing an innovative alternative that offers speed and flexibility (i.e. agile). Despite the conflicting demands inherent in such hybrid approaches, past studies show that they have the potential to adapt, make trade-offs, and drive successful projects (Cao et al. 2009; Fitzgerald et al. 2006; Ramesh et al. 2012).

However, despite the links between ambidexterity and performance (Ramesh et al. 2012), relatively little is known about how companies transition from a purely traditional approach to one that integrates traditional and agile practices together. The trade-offs that are inherent in simultaneously adopting two distinct approaches must be reconciled, but it is unclear if a balance is reached by maintaining a traditional approach with some areas (e.g. planning, requirements gathering) and an agile approach with others (e.g. development, testing) or if all areas of a project can blend together practices from both approaches (e.g. a project subscribes to neither heavy documentation nor minimal documentation, but something in between). In order to examine this practical challenge, we investigate one representative area of systems development projects – knowledge management – that demonstrates clear agile-traditional conflicts. Although this examination alone cannot paint the entire picture of projects transitioning to a hybrid approach, it represents an important step in enhancing our understanding of how firms incrementally adopt agile.

### ***Managing Knowledge in Systems Development Projects***

Knowledge management refers to the process of capturing, storing, sharing, and using knowledge (Davenport and Prusak 1998). Within the systems development process, aspects of knowledge management have been considered relative to how teams elicit requirements, design and develop software, and train users within both agile and traditional development approaches (Dingsoyr et al. 2012; Lyytinen and Robey 1999; Ramesh and Tiwana 1999). Whereas traditional approaches largely focus on knowledge as an ‘object’ that can be documented and exchanged in written form (Alavi and Leidner 2001; Zack 1999), an agile approach positions knowledge more as a ‘relationship’ that is created and transferred through social interaction (Dyba and Dingsoyr 2008; Melnik and Maurer 2004; Nerur and Balijepally 2007; Nidumolu et al. 2001).

This object-relationship differentiation between knowledge management practices relative to the traditional and agile development approaches is supported by Chau et al. (2003), who draw a comparison based on eight categories: documentation, capture of requirements and domain knowledge, training, competence management, trust and care, team composition, continuous learning, and knowledge repositories (refer to Table 1). By comparing the practices inherent in each approach, a variety of unique benefits (and drawbacks) arise. For example, the document-intensive traditional approach to knowledge management is advantageous compared to agile in that “it reduces the likelihood of loss of knowledge as a result of knowledge holders leaving the organization” (p. 2). On the other hand, agile practices related to training, such as pair programming, are inexpensive and valuable for all participants, compared to formal (i.e. traditional) training that can be expensive and take participants away from actual project contributions.

Category	Traditional Approach	Agile Approach
Documentation	Extensive documentation: requirements, design, development, etc.	'Just enough' documentation, which may include techniques such as story cards
Requirements and Domain Knowledge	Requirements captured before initiation of design and development; minimal interaction between development team and customers	Active stakeholders and extensive user participation throughout project
Training	Formal, facilitated training sessions	Informal training practices, such as pair programming
Competence Management	Formal status reports, assigned responsibilities based on document ownership	Daily stand-up meetings to discuss progress
Trust and Care	Low reliance on trust, due to stage gate process that mandates review for completeness of project stages	High reliance on trust within the team, built from practices such as collective code ownership, stand-up meetings, collaborative workspaces and pair programming
Team Composition	Role-based teams, such as business analysts, developers, and testers	Cross-functional teams, with team members playing multiple roles throughout the project
Continuous Learning	Post-mortem reviews at the end of project stages or at project completion	Retrospective practices at the end of sprints to review success factors and obstacles
Knowledge Repositories	Heavy reliance on explicit knowledge stored in documents within formal knowledge repositories	Use of lightweight, informal knowledge repositories in either non-digital (e.g. storyboards) or digital (e.g. LeanKit) form

**Table 1. Traditional-Agile Comparison of Knowledge Management Categories**

This section has outlined key aspects of ambidexterity, agile/traditional development approaches, and the distinguishing factors of knowledge management. In the next section, we outline the steps taken for our data collection and analysis.

## Methodology

We adopt a qualitative approach using a single, in-depth case study drawing on extensive interviews with company executives, project team members, and end users. Rather than conducting interviews at a single point in time, the temporal nature of our research question instead required a longitudinal approach. We first discuss the participating organization and the nature of their systems development project and then detail our data collection and analysis.

### Company and Project Context

The participating organization, which we refer to using the pseudonym TechRecruit, is a staffing and consulting company based in the US Northeast. They specialize in employment recruiting for technology-related positions such as systems administrators, systems developers, business analysts, and IT project managers.

The new CRM system was designed to manage the company's core business of matching identified job opportunities at clients to prospective job seekers with the requisite skill sets, as well as integrate with back-office processes such as accounting. The new system would replace an existing CRM that had been in

place for over a decade, but was no longer supported by the vendor and offered limited scalability for the growing company. Although the old system was generally well liked by management and staff, it was seen as lacking up-to-date functionality. A new system was seen as potentially driving back-office efficiencies in terms of billing and reporting, as well as the addition of useful functionality.

The project commenced in early 2012 with the selection of a software vendor from an original pool of 15 options. The selected vendor was a relatively new, unestablished company at the time and was smaller in size and had less experience than the other vendors, but their product was viewed positively by TechRecruit management on the basis that it was highly customizable and had been designed with the staffing industry in mind (rather than a generic product that was tweaked to the specifics of the industry). The application used a private cloud-based approach, whereby the hardware, storage, and infrastructure was managed offsite by the vendor. The software was based on a .NET Microsoft platform using a Google Chrome web-based interface and was closely integrated with Microsoft Outlook.

The initial implementation of the system took place in three stages between October and December 2012 and employed a traditional development approach. However, the system experienced widespread performance problems and functionality limitations immediately after implementation. Although some improvements were made during the first quarter of 2013, a second implementation of the system, referred to as the 're-launch' was subsequently commissioned in June 2013 to address the identified shortcomings. Due to the issues experienced during the first implementation, the re-launch adopted a number of Scrum-based agile techniques. Like many organizations experimenting with agile techniques, the re-launch did not fully adopt all aspects of agile, but TechRecruit chose a selection techniques that they determined would fit the project and the staff. This included the use of story cards, timeboxing, four-week sprints, stand-up meetings, sprint retrospectives, and planning poker. A range of traditional development techniques also remained in place. The IT Project Manager was a Certified ScrumMaster and the company had previously used Scrum for IT infrastructure projects, but the remainder of the project team had limited experience with agile methods at the outset of the project. The re-launched system was rolled out successfully with improved performance and functionality during February 2014.

### ***Data Collection***

Our data was collected over an eighteen month period in conjunction with the timing of original systems implementation and the subsequent re-launch. 21 interviews were conducted during March and April 2013 and an additional 11 interviews during May-November 2014. Both sets of interviews commenced approximately three months following the initial implementation and again three months after the re-launch. The interviews were recorded and a total of 155 pages of transcribed notes were generated.

The data from 32 interviews<sup>1</sup> were collected using a semi-structured interview approach. Participants were selected based on their participation in the CRM project either as company executives (e.g. CEO, CIO), business management and staff (e.g. Accounting Director, Marketing Manager), IT management and staff (e.g. IT Project Manager, IT Director), or end users (e.g. Financial Analyst, Sales Support Specialist).

### ***Data Analysis***

Qualitative data analysis was conducted on the interview transcripts using NVivo. We coded the data into the eight categories identified in Table 1. The first author conducted a trial coding exercise using a sample of the data and the results were discussed with the second author. The remaining data was then coded and the second author reviewed the results. Further discussion on the results was conducted and all disagreements were satisfactorily resolved. A total of 116 interview segments were coded across the eight categories.

We then compared the collected data from the first implementation (i.e. employing a traditional approach) and the re-launched implementation (i.e. using a hybrid, agile-traditional approach) in order to

---

<sup>1</sup> Of the 32 interviews conducted, 2 were comprised of multiple participants and the remainder were single participant discussions. A total of 28 TechRecruit employees participated in the study, as some individuals were interviewed on multiple occasions.

identify patterns in the knowledge management practices of the project. The results from this exercise are presented below.

## Results

In presenting the results of our analysis, Table 2 highlights the data collected pertaining to the first systems implementation (i.e. using a traditional approach), as well as the data associated with the re-launch (i.e. using a hybrid agile-traditional approach). The data focuses on knowledge management elements of the project and representative quotes for each category are listed in **Appendix A**. The last column notes how each knowledge management category differed from the first implementation to the re-launched implementation; namely, if the development techniques remained traditional, became a hybrid mix of agile and traditional, or transitioned to primarily agile.

Category	First Implementation (Traditional)	Re-launched Implementation (Hybrid Agile-Traditional)	How Did KM Elements Change with the Addition of Agile?
Documentation	A variety of documentation was in place to support the project, including detailed process workflows, project plans, and resource estimations. Although a service level agreement was established with the vendor, it was not viewed as being adhered to.	During the re-launch, formal documentation remained important. This included the creation of a project charter document, revised service level agreement, iteratively updated requirements document, and a technical strategy document signed off by the vendor.	This category remains <b>TRADITIONAL</b> due to the heavy importance placed on formal documentation.
Requirements and Domain Knowledge	An extensive evaluation of system requirements was conducted. Project team members demonstrated a thorough understanding of why the system was being implemented and what it needed to do. Although many requirements were met when the system went live, interviewees identified a variety of significant shortcomings (e.g. performance, reliability).	The re-launch transitioned away from formal ownership of particular deliverables and increasingly relied on cross-functional discussions and decision making. This includes the adoption of a timebox approach, whereby requirements were prioritized and delivered in four week sprints.	This category transitioned to become predominantly <b>AGILE</b> due to additional cross-functional consultation on requirements, alongside a timebox approach to requirements delivery.
Training	Highly structured, lecture-based, classroom training was used in the initial implementation. This approach was commonly viewed as valuable by employees, but it was seen to take away from the day-to-day responsibilities of employees, who commonly work on commission. Training team turnover also negatively impacted the training quality.	The re-launch continued to employ interactive, hands-on training, supplemented by knowledge repositories. The training was derived from materials created for the first implementation, but about half was updated with the assistance of the IT department. The training was slightly more flexible, by allowing users more opportunities to experiment with the system features.	The Training category remains largely <b>TRADITIONAL</b> , due to a continued reliance on formal training.

Competence Management	There was careful consideration taken during the vendor selection process, during data migration, and testing, but the most significant issue was with the vendor, which was viewed by interviewees as understaffed and unresponsive. Formal contracts and responsibilities were established with the vendor to control the deliverable quality.	The re-launch utilized more careful monitoring of issues and more frequent meetings with the vendor. The client took the initiative to more thoroughly test code, independent of vendor testing. A technical strategy document was signed off by vendor and more rigorously defined quality criteria. Stand-up meetings improved the IT Project Manager's ability to address issues as they arose and find solutions.	This category transitioned to a <b>HYBRID</b> approach, drawing on both formal documentation (strategy document defining quality criteria), as well as more informal approaches such as stand-up meetings with project team members.
Trust and Care	A hard deadline was established for the system go-live; however, interviewees disagreed on whether the system was ready to be implemented or if the date should have been pushed back. This appears to have created a level of distrust between executives and staff. Part of this issue stemmed from the lack of a clear business owner/champion for the project. As well, trust between the vendor and the company was strained. Interviewees perceived the vendor as not delivering the product that they had promised.	The re-launch demonstrated a more effective approach by the vendor to identify and resolve defects and deliver on needed functionality. Although the vendor was responsible for much of the coding, the company had extensive access to the source code to make data- and configuration-related changes to the system.  Within the company's project team, tasks were prioritized and the level of effort was estimated with poker cards and team members were given autonomy to sign up for their choice of tasks.	Both traditional approaches (e.g. duplicate testing due to lack of trust in vendor competencies) and agile approaches (e.g. planning poker) to trust and care were demonstrated, indicating this category was adapted to represent a <b>HYBRID</b> approach.
Team Composition	The project team was centralized and included representatives from across the organization. Subject matter experts participated as 'pilot team' members during requirements gathering and testing.	The re-launch restructured the project team into a more decentralized model. Team leaders were set up to oversee staffing deliverables, finance deliverables, and IT deliverables, with a ScrumMaster overseeing the project activities and backlog. This group ran as an agile team, but each team leader independently managed the activities related to their deliverable area.	The team composition continued to draw on functional area knowledge of team members, but became increasingly cross-functional due to the introduction of integrated meetings and a consolidated project backlog. Overall, this category is considered <b>HYBRID</b> .
Continuous Learning	Due to the initial performance and reporting issues with the system, users	Monthly sprint retrospectives were introduced for project team members. Discussions	Continuous learning became distinctly <b>AGILE</b> during the

	became increasingly familiar with the system functions through trial and error in the months following go-live.	included what went well, issues that occurred, and opportunities for improvement.  Team members became increasingly comfortable with agile techniques, such as planning poker.	re-launch, due to the introduction of sprint retrospectives and the increasing comfort the team had with agile techniques.
Knowledge Repositories	Microsoft SharePoint was in place to store system related documents, but usage declined over the course of the project. An issue ticketing system was also in place, to keep track of ongoing bugs and resolutions related to the CRM.	The re-launch employed a more extensive, traceable repository of project knowledge thorough the use of the lightweight agile tool LeanKit. This included story cards and the project backlog. The tool was extensively used throughout the re-launch.	Due to the introduction of the LeanKit tool as a more lightweight alternative to SharePoint, this category transitioned to become increasingly <b>AGILE</b> .

**Table 2. Knowledge Management Characteristics and Changes**

In this section we identified the key characteristics of the eight categories of knowledge management within the first implementation and subsequent re-launch of the CRM system. We also note the extent of the changes to the development techniques resulting from the adoption of a hybrid agile-traditional approach: whether the category remains traditional, if both traditional and agile techniques are employed together in the category (i.e. hybrid), or if the category has evolved to a primarily agile approach. In the next section, we discuss the findings in light of past literature and the implications of the study.

## Discussion

The results noted above provide a view into TechRecruit's transition from a purely traditional development approach to one that simultaneously integrated both traditional and agile techniques. By all accounts, the re-launched implementation was viewed as more successful than the initial implementation and interviewees perceived the introduction of agile techniques as a key factor driving this improved performance. In considering the differences between the two phases of the project, we can gain insights into the steps companies take in transitioning to a hybrid agile-traditional approach and the resulting consequences on how knowledge is managed.

Across the eight knowledge management categories, two remained primarily traditional in nature, three shifted to hybrid (i.e. both agile and traditional), and three progressed to become primarily agile. Although the knowledge management elements of the overall project are broadly seen to be hybrid, these findings highlight the unique configuration of agile and traditional techniques that contribute to this outcome. That is, a hybrid approach can incorporate a mix of distinctly agile or distinctly traditional techniques in particular areas, such as documentation, as well as a collection of techniques that are simultaneously agile *and* traditional. This variability suggests that different paths exist to achieving ambidexterity in hybrid development approaches, which is consistent with past literature (Gibson and Birkinshaw 2004). Because of the structural and cultural challenges presented to companies shifting from a purely traditional to a purely agile approach, our data suggest that the challenges may be eased with incremental adoptions that select techniques that are easiest to implement, are perceived to be acceptable to staff, and stand to address the problem areas within a traditional approach. For example, in the first implementation of the CRM system, many required functions that were delayed due to staff shortages or were not sufficiently tested prior to going live. By adopting four-week sprints and a timebox approach in the re-launch, the new agile techniques were seen as being more effective at delivering on expectations, but were not so revolutionary as to create staff resistance.

Past research also suggests that contextual ambidexterity is achieved not only by attempting to reconcile conflicting objectives, but by encouraging employees to make their own judgments on how this balancing should occur (Gibson and Birkinshaw 2004). However, our results suggest that this can only occur when



employees are sufficiently familiar with the full range of options available; in this case, the traditional and agile techniques. However, because the IT project manager was the only individual on the project team with extensive agile experience, we would expect that future projects at TechRecruit employing a hybrid approach would better achieve benefits from ambidexterity, as staff become more aware of the range of possible options.

One of the key knowledge management implications for companies pursuing an ambidextrous approach is the shifting balance of power from management to development team members that is associated with agile techniques (Nerur and Balijepally 2007; Nerur et al. 2005). Such a shift could be in conflict with cultural norms and generate resistance from management. However, in situations where the traditional approach is producing less than ideal results – as it was at TechRecruit – the transition towards ambidexterity may be made more easily. We found little evidence to support the assertion that managers viewed agile as a threat to their control of knowledge on the project. Rather, they saw the LeanKit tool, story cards, and project poker as an innovative means to create and transfer knowledge that would benefit the project and the company.

Vinekar et al. (2006) outline a model of traditional and agile co-existence that draws on the concept of structural ambidexterity. Such organizations would employ two independent sub-units, one with high exploitive ability that adopts a traditional approach and one with high explorative ability that adopts an agile approach. From a practical perspective, this option avoids at least some of the trade-offs that arise within a hybrid approach. However, mid-sized companies like TechRecruit typically do not have the resources to employ two independent systems development teams. Furthermore, current literature is unclear in concluding on the relative benefits of using a ‘pure’ approach, with recent studies citing the benefits of employing a tailored approach (Cao et al. 2009; Fitzgerald et al. 2006).

## **Conclusions**

This research examines how knowledge management practices are adapted within systems development projects that transition from an exclusive use of traditional practices to also include agile practices. Using a single, longitudinal case study, we draw on interview data from an initial CRM implementation that used a traditional approach, as well as a subsequent re-launch that used a hybrid agile-traditional approach. We find that some knowledge management elements remained traditional, others transitioned to be characteristic of agile, and some integrated both approaches together in a hybrid technique. Our study aids practitioners by providing insights into the opportunities and pitfalls of managing knowledge within hybrid agile-traditional development projects. For researchers, this paper applies the concept of ambidexterity in an agile development context as it relates to knowledge management. Our findings extend the literature on the incremental trade-offs that companies make in attempting to simultaneously explore and exploit two development approaches.

As with any study, our work has limitations, as well as opportunities for further research. First, we focus primarily on knowledge management elements of the development project and it is unclear how our findings may generalize to other areas. Future research could consider the trade-offs between traditional and agile approaches in topics such as planning, development, or testing. Second, the company participating in our study had limited prior experience with agile, which could differentiate our results from a company with a more mature hybrid approach. By considering a wider range of companies employing a hybrid approach, future research could identify if hybrid expertise could link to the achievement of ambidexterity competencies. Finally, we focused on two phases of a single CRM implementation, whereas subsequent projects may employ a different knowledge management approach using a distinct collection of development techniques. Future research may benefit from more extended longitudinal examinations of the evolution of agile adoption in organizations, as it could shed light on the complex process of incrementally adopting an agile approach over time.

## **REFERENCES**

- Alavi, M., and Leidner, D.E. 2001. "Knowledge Management and Knowledge Management Systems: Conceptual Foundations and Research Issues," *MIS Quarterly* (25:1), pp. 107-136.

- Beck, K., Beedle, M., Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., and Marick, B. 2001. "Manifesto for Agile Software Development." Retrieved August 2014, from <http://agilemanifesto.org>
- Boehm, B., and Turner, R. 2003. "Using Risk to Balance Agile and Plan-Driven Methods," *Computer* (36:6), pp. 57-66.
- Cabral, A.Y., Ribeiro, M.B., Lemke, A.P., Silva, M.T., Cristal, M., and Franco., C. 2009. "A Case Study of Knowledge Management Usage in Agile Software Projects," *Lecture Notes in Business Information Processing* (24), pp. 627-638.
- Cao, L., Mohan, K., Xu, P., and Ramesh, B. 2009. "A Framework for Adapting Agile Development Methodologies," *European Journal of Information Systems* (18:4), pp. 332-343.
- Chau, T., Maurer, F., and Melnik, G. 2003. "Knowledge Sharing: Agile Methods Vs. Tayloristic Methods," *Twelfth IEEE International Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises*.
- Davenport, T.H., and Prusak, L. 1998. *Working Knowledge: How Organizations Manage What They Know*. Boston: Harvard Business School.
- Dingsoyr, T., Nerur, S., Balijepally, V., and Moe, N.B. 2012. "A Decade of Agile Methodologies: Towards Explaining Agile Software Development," *Journal of Systems and Software* (85:6), pp. 1213-1221.
- Dyba, T., and Dingsoyr, T. 2008. "Empirical Studies of Agile Software Development: A Systematic Review," *Information and Software Technology* (50:9-10), pp. 833-859.
- Fitzgerald, B., Hartnett, G., and Conboy, K. 2006. "Customising Agile Methods to Software Practices at Intel Shannon," *European Journal of Information Systems* (15:2), pp. 200-213.
- Gibson, C.B., and Birkinshaw, J. 2004. "The Antecedents, Consequences, and Mediating Role of Organizational Ambidexterity," *Academy of Management Journal* (47:2), pp. 209-226.
- Li, L., Zhang, W., and Mao, J.-Y. 2013. "The Business Value of Crm Technology: From the Perspective of Organizational Ambidexterity," in: *Nineteenth Americas Conference on Information Systems*. Chicago, Illinois.
- Lyytinen, K., and Robey, D. 1999. "Learning Failure in Information Systems Development," *Information Systems Journal* (9:2), pp. 85-101.
- Marabelli, M., Frigerio, C., and Rajola, F. 2012. "Ambidexterity in Service Organizations: Reference Models from the Banking Industry," *Industry and Innovation* (19:2), pp. 109-126.
- Melnik, G., and Maurer, F. 2004. "Direct Verbal Communication as a Catalyst of Agile Knowledge Sharing," in: *Agile Development Conference*. pp. 21-31.
- Nerur, S., and Balijepally, V. 2007. "Theoretical Reflections on Agile Development Methodologies," *Communications of the ACM* (50:3), pp. 79-83.
- Nerur, S., Mahapatra, R., and Mangalaraj, G. 2005. "Challenges of Migrating to Agile Methodologies," *Communications of the ACM* (48:5), pp. 73-78.
- Nidumolu, S.R., Subramani, M., and Aldrich, A. 2001. "Situated Learning and the Situated Knowledge Web: Exploring the Ground beneath Knowledge Management," *Journal of Management Information Systems* (18:1), pp. 115-150.
- Polanyi, M. 1962. *Personal Knowledge: Towards a Post-Critical Philosophy*. New York: Harper Torchbooks.

- Port, D., and Bui, T. 2009. "Simulating Mixed Agile and Plan-Based Requirements Prioritization Strategies: Proof-of-Concept and Practical Implications," *European Journal of Information Systems* (18:4), pp. 317-331.
- Raisch, S., Birkinshaw, J., Probst, G., and Tushman, M.L. 2009. "Organizational Ambidexterity: Balancing Exploitation and Exploration for Sustained Performance," *Organization Science* (20:4), pp. 685-695.
- Ramesh, B., Mohan, K., and Cao, L. 2012. "Ambidexterity in Agile Distributed Development: An Empirical Investigation," *Information Systems Research* (23:2), pp. 323-339.
- Ramesh, B., and Tiwana, A. 1999. "Supporting Collaborative Process Knowledge Management in New Product Development Teams," *Decision Support Systems* (27:1-2), pp. 213-235.
- Rothaermel, F.T., and Alexandre, M.T. 2009. "Ambidexterity in Technology Sourcing: The Moderating Role of Absorptive Capacity," *Organization Science* (20:4), pp. 759-780.
- Stoetzel, M., and Wiener, M. 2013. "Challenges and Dilemmas in Open Innovation: Ambidexterity as Management Approach," in: *11th International Conference on Wirtschaftsinformatik*. Leipzig, Germany.
- Tiwana, A. 2010. "Systems Development Ambidexterity: Explaining the Complementary and Substitutive Roles of Formal and Informal Controls," *Journal of Management Information Systems* (27:2), pp. 87-126.
- Turner, N., Swart, J., and Maylor, H. 2013. "Mechanisms for Managing Ambidexterity: A Review and Research Agenda," *International Journal of Management Reviews* (15), pp. 317-332.
- Tushman, M.L., and O'Reilly, C.A. 1996. "Ambidextrous Organizations: Managing Evolutionary and Revolutionary Change," *California Management Review* (38:4), pp. 8-30.
- Vinekar, V., Slinkman, C.W., and Nerur, S. 2006. "Can Agile and Traditional Systems Development Approaches Coexist? An Ambidextrous View," *Information Systems Management* (23:3), pp. 31-42.
- Zack, M.H. 1999. "Developing a Knowledge Strategy," *California Management Review* (41:3), pp. 125-145.

## APPENDIX A – Representative Interviewee Quotes

Category	First Implementation (Traditional)	Re-launched Implementation (Hybrid Agile-Traditional)
Documentation	<i>'[The IT Project Manager] documented ...the workflows, the dashboard, the set-up, the reports. We need more time to document the workflows of Finance and Accounting'. CIO</i>	<i>'It would be email updates as well. So during that process that might take us three to four weeks during that to do all those things. We'd have weekly meetings and then after the meeting there'd be an email with what was discussed in that meeting and the requirements doc would be updated and then the next meeting we'd go in and discuss it and make sure.' VP Recruiting</i>
Requirements and Domain Knowledge	<i>'I'm interested in data. How many clients you call on today? How many candidates you reach out? How many resumes have you read? Data are the</i>	<i>'The essence [of agile development] is really the culture, the fit, getting people's head around that and you might find some people who although they are technically</i>

	<p>output of their efforts. And I need their data to determine where we can be better and more efficient; where maybe we miss opportunities. I need data to value what we do better and what we don't.' Chief Marketing Officer</p>	<p>very talented, they're just unable to come to grips with working and being willing to share and be willing to extend themselves...the idea about agile is even though I'm a developer, I should be able to write functional tasks...if I'm a business development guy I should be able to do some UI work. It might be that I'm not a UI expert but now gaining some appreciation of what it takes to basically implement the user and integrate that with the business layer....So we've got that in terms of how to do that more effectively.' CIO</p>
Training	<p>'The big challenge now is training people on [the new CRM system]. We have different people geographically located and also they have to take time to train getting away from their phone... they lose money! So what are you are going to do?...These guys are competitive, they are awesome. But they don't want to sit in a classroom and learn. The challenge is, how can we train? How to teach them? I need to be creative.' Training Specialist</p>	<p>We put in exercises, like formal step one, try this, step two, do this, step three, do that. We'd have 10-15 steps or whatever on an exercise. But after I got done teaching I would tell them, 'listen, here's an exercise' and I would keep it showing on the projector but I would say, 'if you want to try something different because you want to see if it does this or does that you know, please feel free to go away from the procedure, you don't have to follow it step by step and letter by letter.' Training Consultant</p>
Competence Management	<p>'The vendor for me was a big challenge. We had to manage a lot of people: our core team, internal IT Team, and the [vendor], who didn't have a Project Manager. The vendor is very small and they didn't have the resources that they originally planned. So externally the problem is to complete development with the vendor and keep on track of their deliverables. But, they are understaffed and you can't control it because they don't work for us.' IT Project Manager</p>	<p>'We ended up buying a monitoring tool and implementing that just to watch and see and set up a baseline of certain things to happen within the guidelines of 0-2 seconds, 3-5 seconds and 5 seconds or longer. And anything that takes 5 seconds or longer would get flagged and we'd go and look at the processes that are running and such and share that information back and forth with [the vendor]. VP Recruiting</p>
Trust and Care	<p>'This project was bigger than we thought and there were not enough people working on it. People worked too many hours and they couldn't follow through on everything. Executives underestimated the impact that this change would have on the company. Why did Executives [decide to] go live?...The system was not ready to go-live for the back office, we needed more time and a better plan of attack.' Accounting Director</p>	<p>Sometimes [the vendor] would agree to make a change and our IT Team does a great job and we have a test environment set up, they want to be able to test it to their liking before they're going to put it into production for us. And every once in a while we'd go to do something and it would lock up. And what happened was [that the vendor] would put something into production and we wouldn't know it. And you know, so there was no communication sometimes that way...There was a lot of times in meetings with them and on the phone and it was like 'You can't just do that to us, you've got to tell us...we want to be able to test it, but if we're not going to test</p>

		<i>it, you've at least got to tell us that you did it'. VP Recruiting</i>
Team Composition	<i>'We have a cross-functional team. This is the way we collaborate, organize into groups, meet very regularly, and communicate very regularly. So there is collaboration between all the parts of the company. Chief Marketing Officer</i>	<i>'So an example, I was the project manager for IT...so let's say one of the IT deliverables would be we had to create 15 laptops with a localized email environment and other configurations on the laptop specific for training purposes. So what I would then do is I would negotiate with my infrastructure team to get them to agree to a deliverable where I would actually set the expectations around requirements and what needed to get done. And then I would basically come back to [the Scrum Master] and say 'Okay, I'm committing for this sprint to get this effort done and it will be done within this four-week period of time'. So although the team who was acting on it wasn't really part of the Agile [team], I was part of the Agile team and then I worked to extend myself and then to basically commit to doing what I needed to get done.' CIO</i>
Continuous Learning	<i>'[Business representatives have] meetings with the IT people every week. In these meetings, [the trainer] talks about issues and things that are updated, things that are changed in the system. We have to still work on the performance, workflows, re-calculator, search features.' Business Manager</i>	<i>'We also did retrospectives about what we ran into in terms of issues, what we did well, what we didn't do well and why. There were issues we ran into and then we picked off one or two things that we thought we had to do better about.' CIO</i>
Knowledge Repositories	<i>They've had a SharePoint system for a long time, it's a document management system but nobody uses it, nobody used it. And it's a very difficult system to find documents in because one of the interfaces for it is just to put in folders and files similar to Windows systems where they would just have to guess which path to go down, look in this folder then look in this folder then look in this folder and keep on following the path until they discovered, 'okay, it's not here'. Go down a different path to try to find something. Training Consultant</i>	<i>'We looked into a couple software options to help streamline that process and...I came up with LeanKit because they had the whiteboard with the sticky notes, you could assign tasks to people, I could schedule certain things, we had pretty good analytics to see tasks being completed. What's reasonable for how many hours are allocated to each person and then I could track things that came up during our sprints that needed...I'd put them to a parking lot as they called it, because it needed to be vetted out further outside of our daily scrum.' IT Project Manager</i>